IOWA STATE UNIVERSITY
Data Storage Lab

FAST'22
20th USENIX Conference on File and Storage Technologies
FEBRUARY 22–24, 2022
SANTA CLARA, CA, USA
https://www.usenix.org/conference/fast22

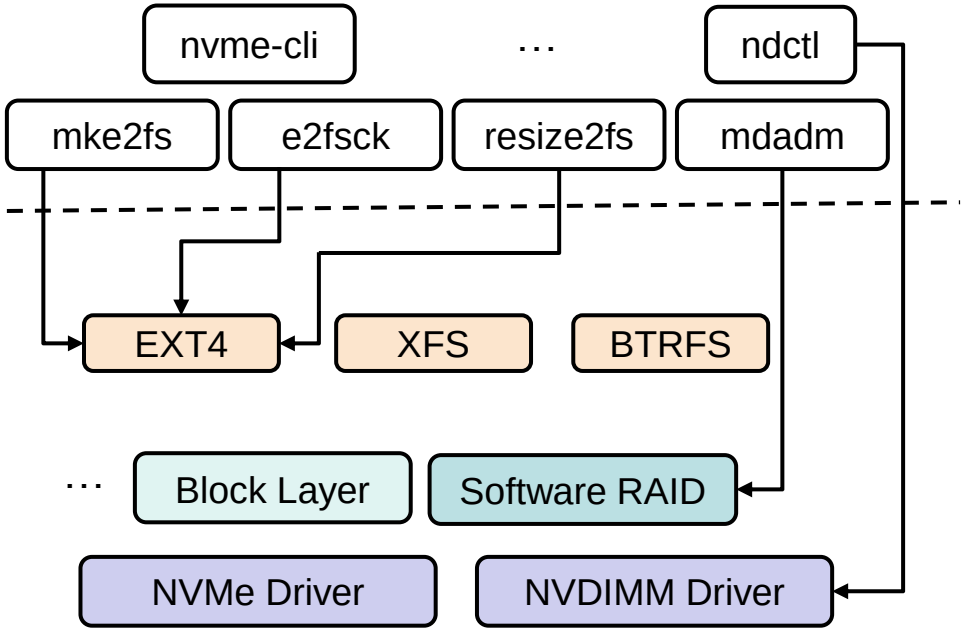
Understanding Configuration Issues in Storage Systems

Tabassum Mahmud, *Mai Zheng*

Department of Electrical and Computer Engineering
Iowa State University

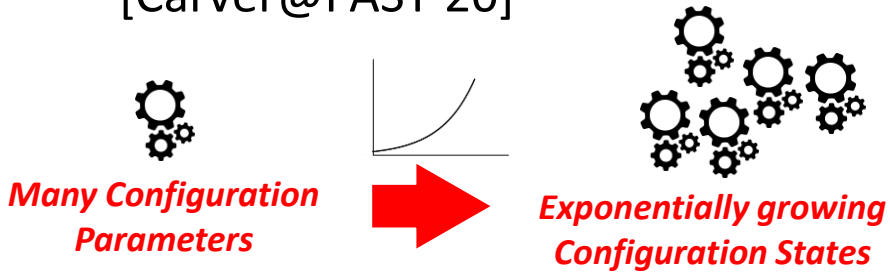
Storage Stack Has Many Configurable Components

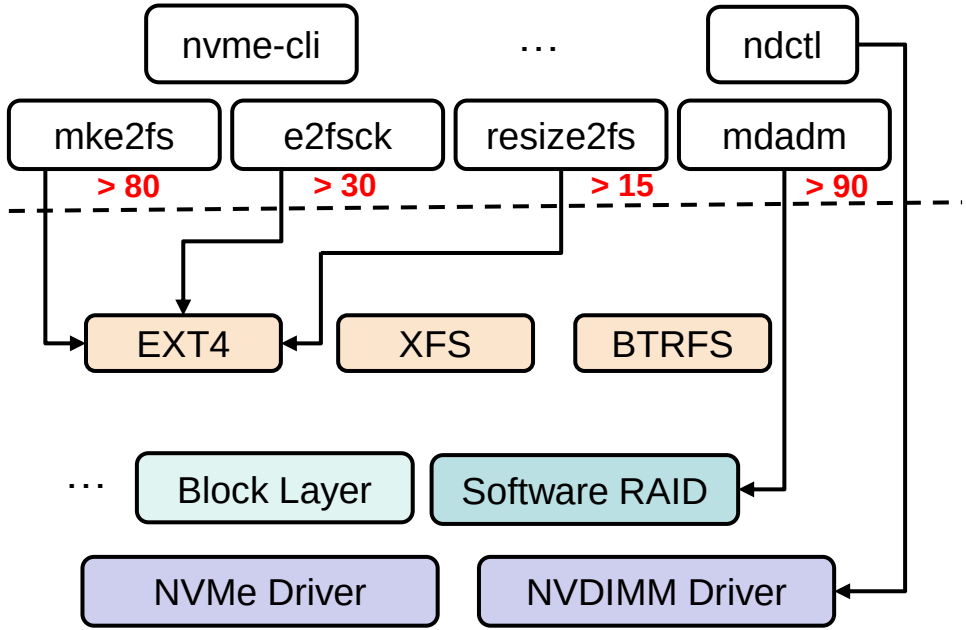
- User-level Programs
  - E.g., e2fsck, resize2fs, mdadm, ..
- Kernel Modules
  - File Systems
    - E.g.: EXT4, XFS, …
  - Software RAID
  - Device Drivers



Each Component has many configuration parameters

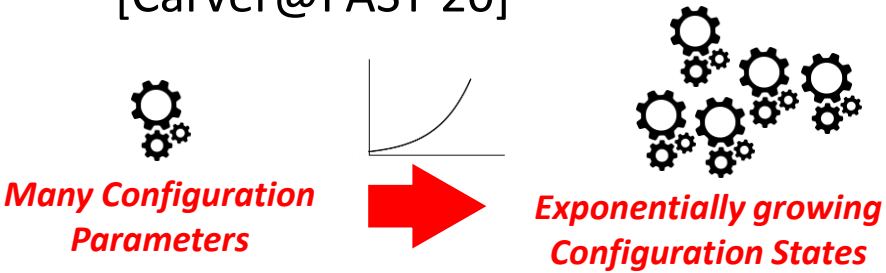
- E.g., *mke2fs* has > 80 configuration parameters with a wide range of values, resulting in over 10^{37} configuration states in EXT4 [Carver@FAST'20]



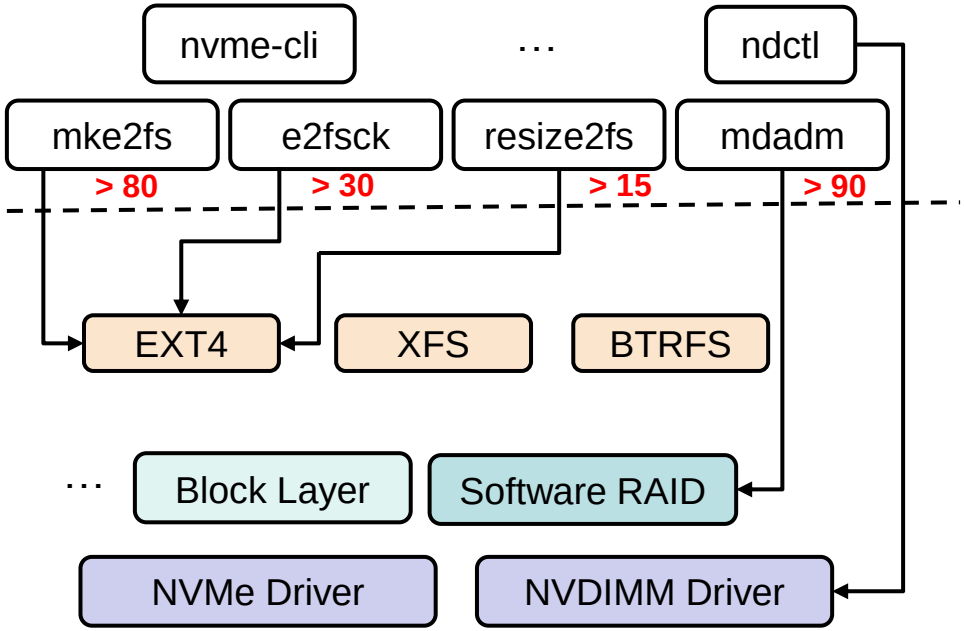


Each Component has many configuration parameters

- E.g., *mke2fs* has > 80 configuration parameters with a wide range of values, resulting in over 10^{37} configuration states in EXT4 [Carver@FAST'20]



Difficult to Exhaust All Possible States



Many Configuration-Related Issues Exist!

- E.g. a bug only exposed when two specific configuration parameters of *e2fsck* are used

```
$./e2fsck –n –E discard
```

```
From f0fe5daecdb0c88afb76c23c77494bbe86e1cd2b Mon Sep 17 00:00:00 2001
From: Lukas Czerner <lczerner@redhat.com>
Date: Sun, 11 Mar 2012 15:35:06 -0400
Subject: [PATCH 3800/6469] e2fsck: do not discard when in read only mode

diff --git a/e2fsck/unix.c b/e2fsck/unix.c
index 6f97b0f2..b31a1e31 100644
--- a/e2fsck/unix.c
+++ b/e2fsck/unix.c
@@ -903,6 +903,11 @@ static errcode_t PRS(int argc, char *argv[], e2fsck_t *ret_ctx)
        profile_set_syntax_err_cb(syntax_err_report);
        profile_init(config_fn, &ctx->profile);

+       /* Turn off discard in read-only mode */
+       if ((ctx->options & E2F_OPT_NO) &&
+           (ctx->options & E2F_OPT_DISCARD))
+               ctx->options &= ~E2F_OPT_DISCARD;
+
        if (flush) {
                fd = open(ctx->filesystem_name, O_RDONLY, 0);
                if (fd < 0) {
```

A configuration-related bug patch

Many Configuration-Related Issues Exist!

- E.g. a bug only exposed when two specific configuration parameters of *e2fsck* are used

read-only mode

```
$./e2fsck -n -E discard
```

discard free blocks

```
From f0fe5daecdb0c88afb76c23c77494bbe86e1cd2b Mon Sep 17 00:00:00 2001
From: Lukas Czerner <lczerner@redhat.com>
Date: Sun, 11 Mar 2012 15:35:06 -0400
Subject: [PATCH 3800/6469] e2fsck: do not discard when in read only mode

diff --git a/e2fsck/unix.c b/e2fsck/unix.c
index 6f97b0f2..b31a1e31 100644
--- a/e2fsck/unix.c
+++ b/e2fsck/unix.c
@@ -903,6 +903,11 @@ static errcode_t PRS(int argc, char *argv[], e2fsck_t *ret_ctx)
        profile_set_syntax_err_cb(syntax_err_report);
        profile_init(config_fn, &ctx->profile);

+       /* Turn off discard in read-only mode */
+       if ((ctx->options & E2F_OPT_NO) &&
+           (ctx->options & E2F_OPT_DISCARD))
+               ctx->options &= ~E2F_OPT_DISCARD;
+
        if (flush) {
                fd = open(ctx->filesystem_name, O_RDONLY, 0);
                if (fd < 0) {
```

A configuration-related bug patch

Many Configuration-Related Issues Exist!

- E.g. a bug only exposed when two specific configuration parameters of *e2fsck* are used

read-only mode

```
$./e2fsck -n -E discard
```

discard free blocks

Expectation:
Make no changes in read-only mode

```
From f0fe5daecdb0c88afb76c23c77494bbe86e1cd2b Mon Sep 17 00:00:00 2001
From: Lukas Czerner <lczerner@redhat.com>
Date: Sun, 11 Mar 2012 15:35:06 -0400
Subject: [PATCH 3800/6469] e2fsck: do not discard when in read only mode

diff --git a/e2fsck/unix.c b/e2fsck/unix.c
index 6f97b0f2..b31a1e31 100644
--- a/e2fsck/unix.c
+++ b/e2fsck/unix.c
@@ -903,6 +903,11 @@ static errcode_t PRS(int argc, char *argv[], e2fsck_t *ret_ctx)
        profile_set_syntax_err_cb(syntax_err_report);
        profile_init(config_fn, &ctx->profile);

+       /* Turn off discard in read-only mode */
+       if ((ctx->options & E2F_OPT_NO) &&
+           (ctx->options & E2F_OPT_DISCARD))
+               ctx->options &= ~E2F_OPT_DISCARD;
+
        if (flush) {
                fd = open(ctx->filesystem_name, O_RDONLY, 0);
                if (fd < 0) {
```

A configuration-related bug patch

Many Configuration-Related Issues Exist!

- E.g. a bug only exposed when two specific configuration parameters of *e2fsck* are used

read-only mode

```
$./e2fsck -n -E discard
```

discard free blocks

Expectation:	***Unexpected Issue:***
Make no changes in read-only mode	*Discards free blocks*

```
From f0fe5daecdb0c88afb76c23c77494bbe86e1cd2b Mon Sep 17 00:00:00 2001
From: Lukas Czerner <lczerner@redhat.com>
Date: Sun, 11 Mar 2012 15:35:06 -0400
Subject: [PATCH 3800/6469] e2fsck: do not discard when in read only mode

diff --git a/e2fsck/unix.c b/e2fsck/unix.c
index 6f97b0f2..b31a1e31 100644
--- a/e2fsck/unix.c
+++ b/e2fsck/unix.c
@@ -903,6 +903,11 @@ static errcode_t PRS(int argc, char *argv[], e2fsck_t *ret_ctx)
        profile_set_syntax_err_cb(syntax_err_report);
        profile_init(config_fn, &ctx->profile);

+       /* Turn off discard in read-only mode */
+       if ((ctx->options & E2F_OPT_NO) &&
+           (ctx->options & E2F_OPT_DISCARD))
+               ctx->options &= ~E2F_OPT_DISCARD;
+
        if (flush) {
                fd = open(ctx->filesystem_name, O_RDONLY, 0);
                if (fd < 0) {
```

A configuration-related bug patch

Limitations of State-of-the-Art

- Practical test suites may miss many configuration states

Test Suite	*Target Software*	*No. of Config. Param.*	*No. of Config. Used*
e2fsprogs	e2fsck	35	6 (17.1%)
e2fsprogs	resize2fs	15	7 (46.7%)
xfstests	EXT4	85	29 (34.1%)
mdadm	mdadm/Software RAID	95	63(39.8%)

Limitations of State-of-the-Art

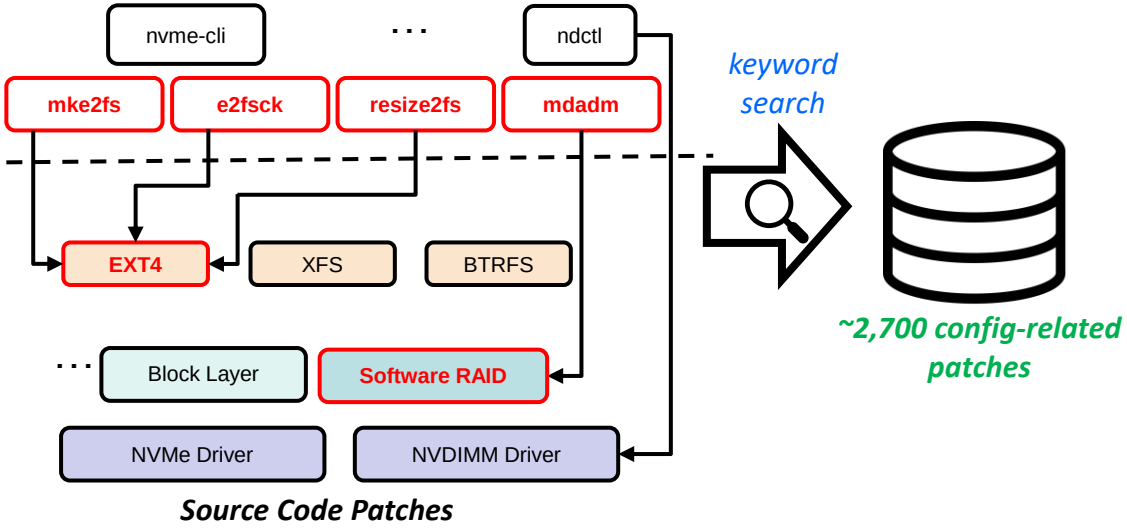
- Practical test suites may miss many configuration states

Test Suite	*Target Software*	*No. of Config. Param.*	*No. of Config. Used*
e2fsprogs	e2fsck	35	6 (17.1%)
e2fsprogs	resize2fs	15	7 (46.7%)
xfstests	EXT4	85	29 (34.1%)
mdadm	mdadm/Software RAID	95	63(39.8%)

How to test configuration-related issues efficiently?

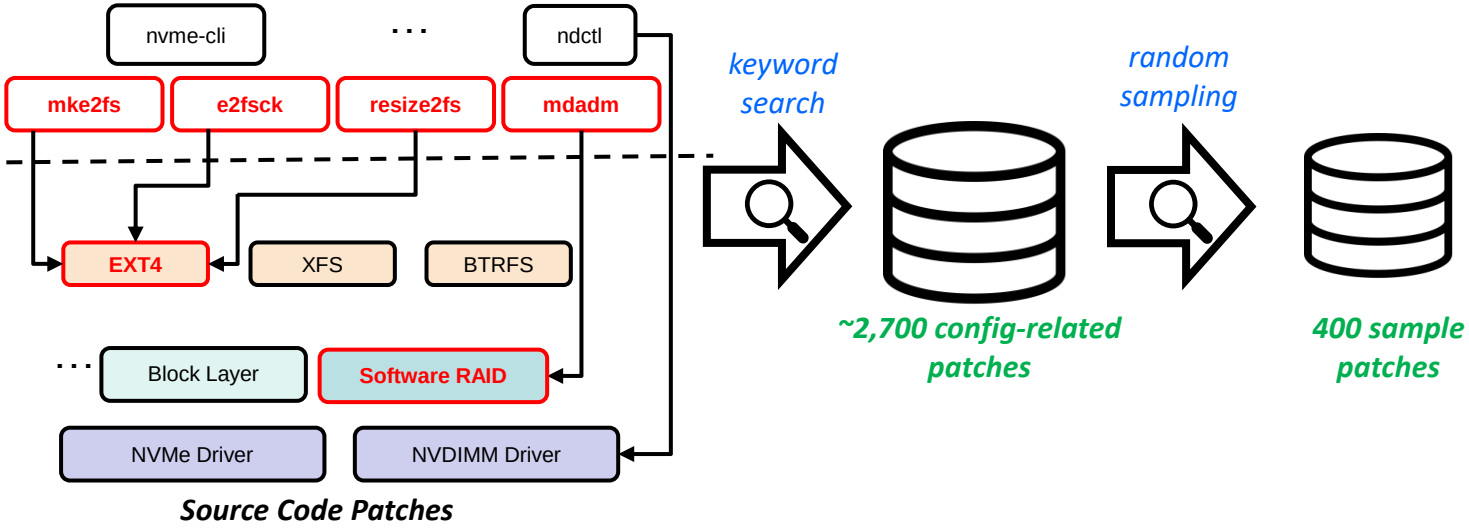
Our Efforts: An Empirical Study

- Analyze configuration-related storage system bugs and relevant source code to understand bug patterns and configuration dependencies



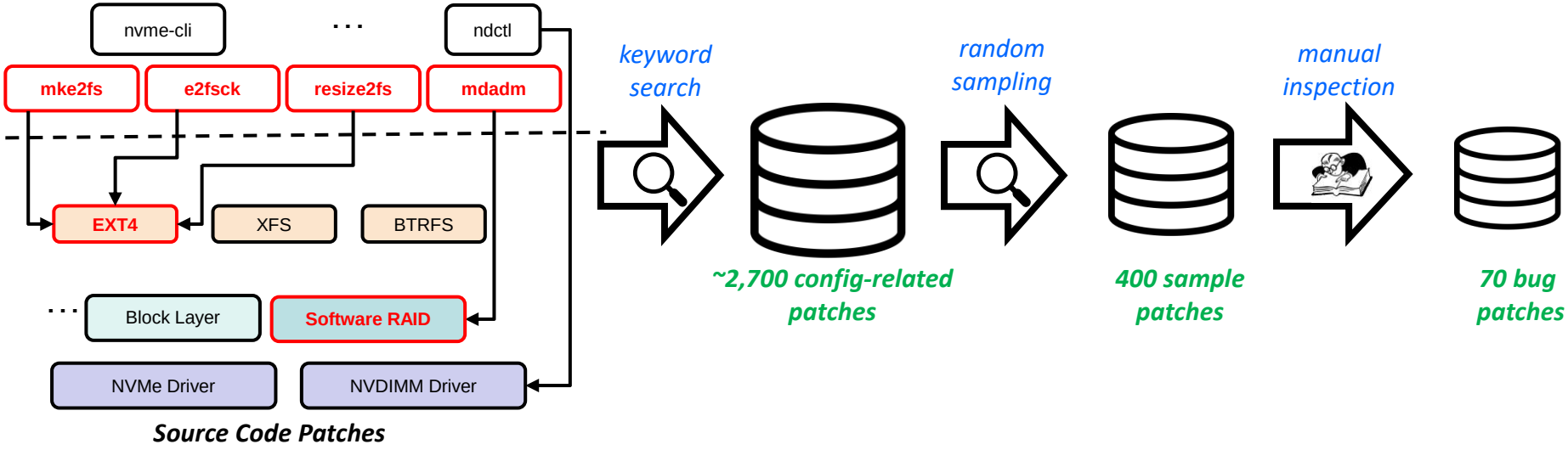
Our Efforts: An Empirical Study

- Analyze configuration-related storage system bugs and relevant source code to understand bug patterns and configuration dependencies



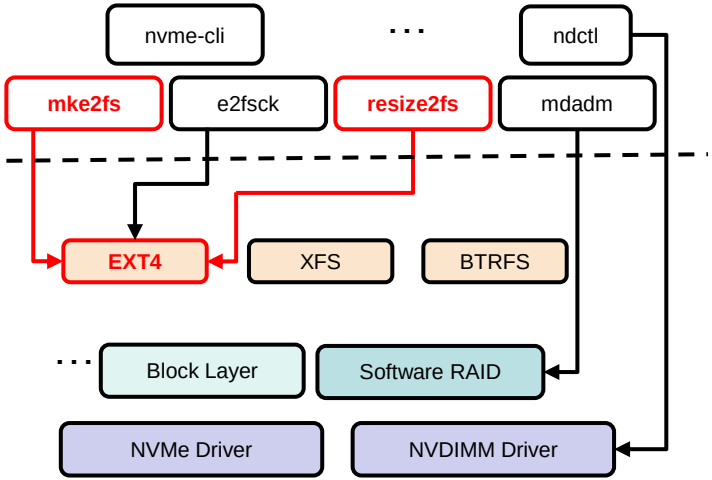
Our Efforts: An Empirical Study

- Analyze configuration-related storage system bugs and relevant source code to understand bug patterns and configuration dependencies



Our Efforts: An Empirical Study

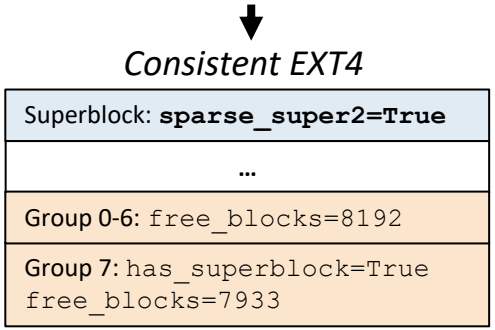
- Key observation #1: many issues involve multiple components!
  - E.g., a bug involving EXT4 configuration parameters and *resize2fs*

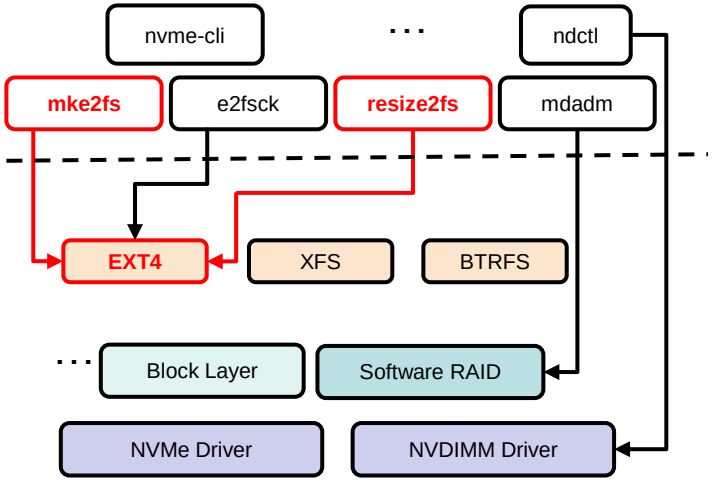


Our Efforts: An Empirical Study

- Key observation #1: many issues involve multiple components!
  - E.g., a bug involving EXT4 configuration parameters and *resize2fs*

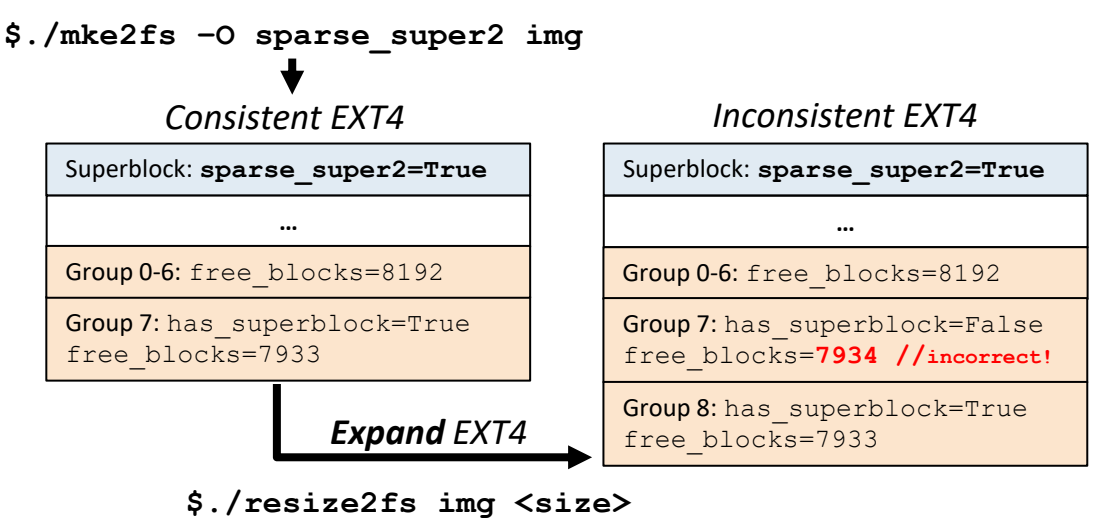
```
$./mke2fs -O sparse_super2 img
```

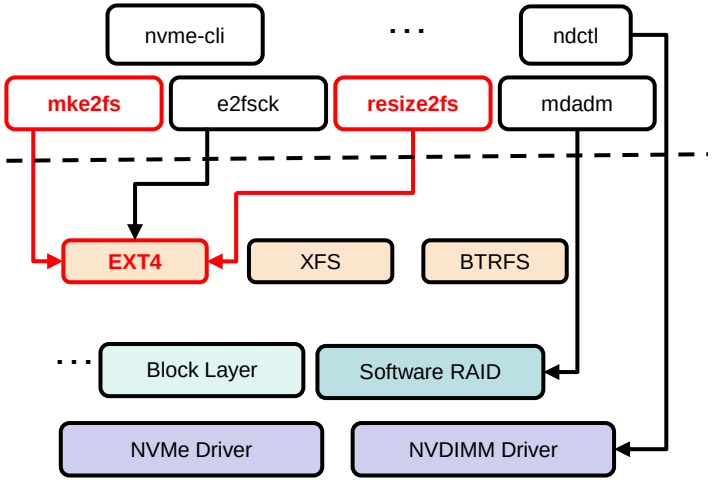




Our Efforts: An Empirical Study

- Key observation #1: many issues involve multiple components!
  - E.g., a bug involving EXT4 configuration parameters and *resize2fs*





Our Efforts: An Empirical Study

- Key observation #2: multi-level configuration dependencies in the source code should be considered to guide configuration testing

```
Example:
$./mke2fs –b 1024 –O sparse_super2, resize_inode, ^meta_bg img
…
$./resize2fs img <size>
```

Our Efforts: An Empirical Study

- Key observation #2: multi-level configuration dependencies in the source code should be considered to guide configuration testing

Example:

```
$./mke2fs -b 1024 -O sparse_super2, resize_inode, ^meta_bg img
…
$./resize2fs img <size>
```

```
PRS(…){
    …
if (b < EXT2_MIN_BLOCK_SIZE ||
        b > EXT2_MAX_BLOCK_SIZE)
            exit(1);

}
```

Individual parameters need to satisfy their own value constraints (Self-Dependency)

Our Efforts: An Empirical Study

- Key observation #2: multi-level configuration dependencies in the source code should be considered to guide configuration testing

Example:

```
$./mke2fs –b 1024 –O sparse_super2, resize_inode, ^meta_bg img
…
$./resize2fs img <size>
```

```
if (ext2fs_has_feature_meta_bg(&fs_param) &&
     ext2fs_has_feature_resize_inode
          (&fs_param)){
                    exit(1);
}
```

Multiple parameters may need to satisfy relative constraints within one program (Cross-Parameter Dependency)

Our Efforts: An Empirical Study

- Key observation #2: multi-level configuration dependencies in the source code should be considered to guide configuration testing

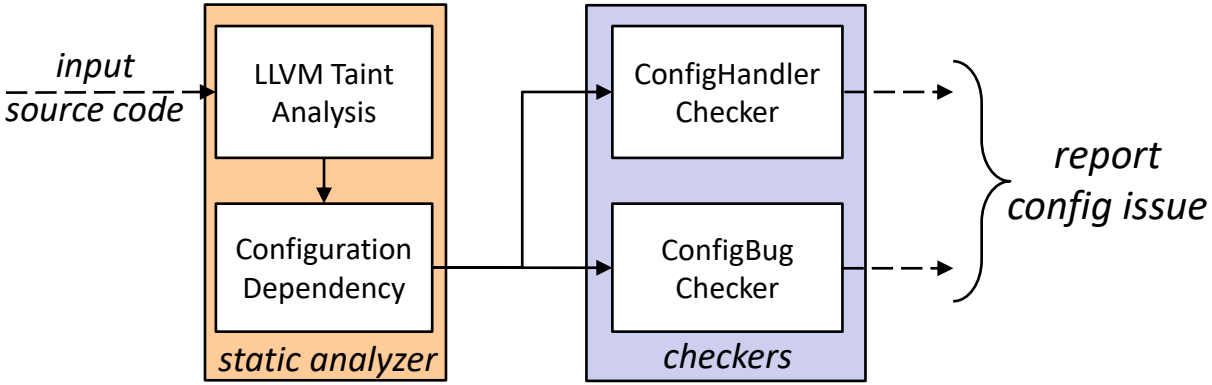
Example:
$./mke2fs -b 1024 -O sparse_super2, resize_inod
…
$./resize2fs img <size>

```
resize_fs(…){
    …
    ret = clear_sparse_super2_last_group(…)
}
clear_sparse_super2(…){
    …
  if(!extfs_has_feature_sparse_super2)
        return 0;
  if(last_bg <= old_last_bg)
        return 0;
    // execute critical functionality
}
```

Multiple parameters may need to satisfy relative constraints across programs (Cross-Program Dependency)

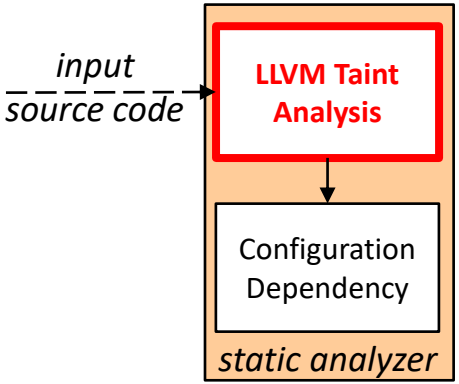
Our Efforts: A Configuration-Issue Analyzer

- Two major modules
  - Static Analyzer: derive configuration dependencies to prioritize configuration states for testing
  - Checkers: check different configuration-related issues based on dependencies



Our Efforts: A Configuration-Issue Analyzer

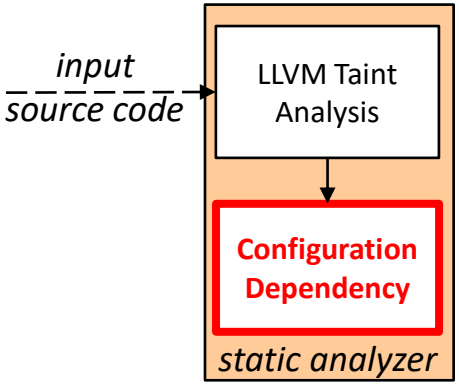
- Two major modules
  - Static Analyzer: derive configuration dependencies to prioritize configuration states for testing
  - Checkers: check different configuration-related issues based on dependencies



Generate taint traces for each parameter and establish relationships between parameters

Our Efforts: A Configuration-Issue Analyzer

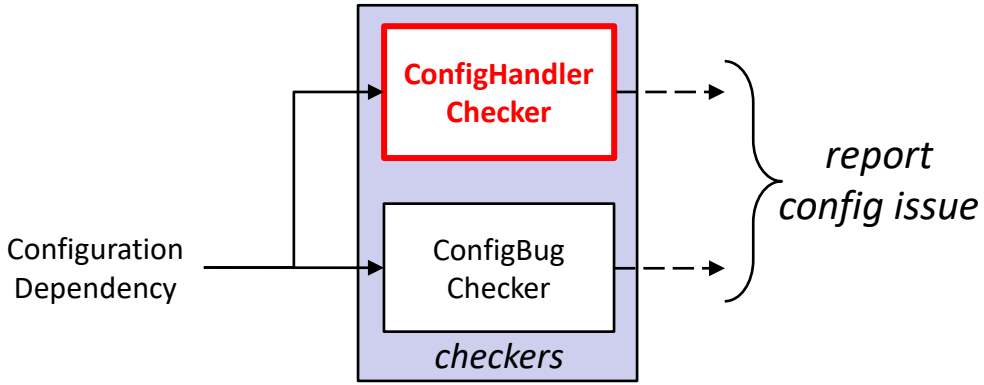
- Two major modules
  - Static Analyzer: derive configuration dependencies to prioritize configuration states for testing
  - Checkers: check different configuration-related issues based on dependencies



Analyzed component	Self dependency	Cross-parameter	Cross-program
mke2fs	17	3	-
e2fsck	2	-	2
resize2fs	5	-	3

Our Efforts: A Configuration-Issue Analyzer

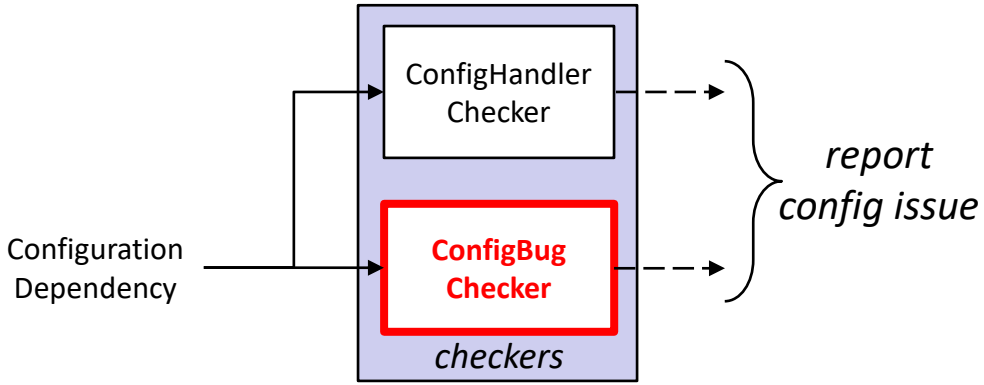
- Two major modules
  - Static Analyzer: derive configuration dependencies to prioritize configuration states for testing
  - Checkers: check different configuration-related issues based on dependencies



Violate the configuration dependencies to test handling of wrong configuration

Our Efforts: A Configuration-Issue Analyzer

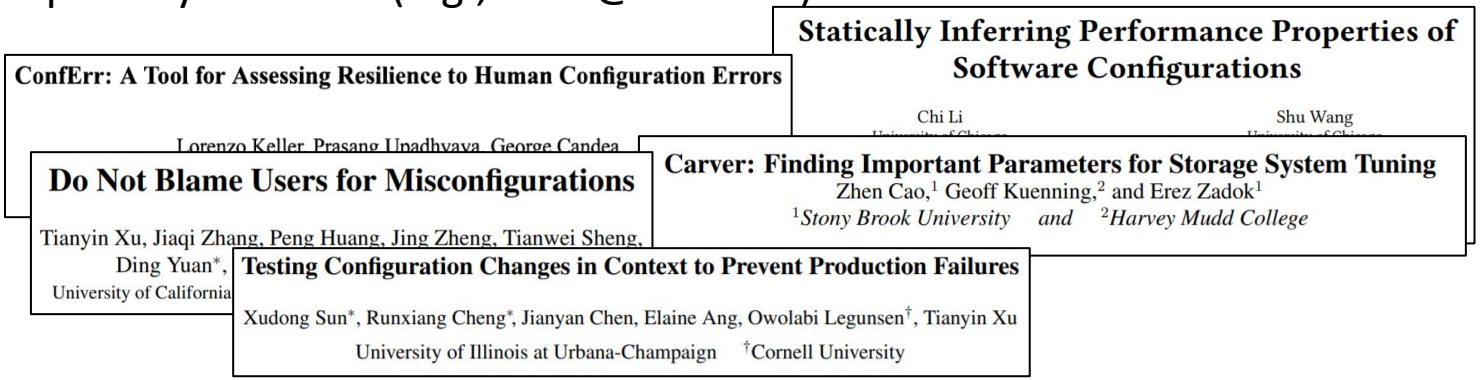
- Two major modules
  - Static Analyzer: derive configuration dependencies to prioritize configuration states for testing
  - Checkers: check different configuration-related issues based on dependencies



Follow the configuration dependencies to dive deep into the source code for testing

Related Work

- Suboptimal for addressing configuration related issues in storage stack
  - Focus on one single application (e.g., SPEX@SOSP'13, ConErr@DSN'08, ctest@OSDI'20)
  - Focus on performance related configurations (e.g., Carver@FAST'20)
  - Rely on profiling traces (e.g., Carver@FAST'20)
  - Focus on Java Programs (e.g., LearnConf@Eurosys'20)
  - Not publicly available (e.g., SPEX@SOSP'13)



Conclusion & Future work

- An empirical study and preliminary prototype
  - Many cases involve multiple components
  - Static analysis can be used to derive the dependencies from source code
  - A static analysis based prototype leveraging multi-level dependencies for effective testing
- Next Steps
  - Fully implement prototype and minimize manual annotation
  - Apply and evaluate the prototype on different storage systems including file systems, software RAID, databases, etc.

Conclusion & Future work

- An empirical study and preliminary prototype
  - Many cases involve multiple components
  - Static analysis can be used to derive the dependencies from source code
  - A static analysis based prototype leveraging multi-level dependencies for effective testing
- Next Steps
  - Fully implement prototype and minimize manual annotation
  - Apply and evaluate the prototype on different storage systems including file systems, software RAID, databases, etc.

Thank You!